**República Bolivariana de Venezuela**

**Ministerio del Poder Popular para la Educación Universitaria**

**U.P.T.P “Juan de Jesús Montilla”**

**Acarigua. Edo – Portuguesa**

**Estudiantes:**

**Gallardo Mariana C.I: V-24.587.862**

**Marcano Daniel C.I.: V-8.283.754**

**Colmenárez Gregorio C.I.: V-21394.280**

**Uzcátegui José C.I.: V-20.607.984**

**Martínez Alejandro C.I.: V-18.291.990**

**Paniccia Leonardo C.I.: V-21.057.756**

**Ing. Informática**

**Sección: 437**

**Prof: Ing. Yanmar Pereira**

**Acarigua, Noviembre 2014**

**Introducción en función del objetivo principal de construir una base de datos**

**1.Modelo**

Un modelo es un conjunto de herramientas conceptuales para describir datos, sus relaciones, su significado y sus restricciones de consistencia.

**Características**

* Es el proceso de analizar los aspectos de interés para una organización y la relación que tienen unos con otros.
* Resulta en el descubrimiento y documentación de los recursos de datos del negocio.
* El modelado hace la pregunta “¿Qué? " en lugar de "¿Cómo? ", ésta última orientada al procesamiento de los datos.
* Es una tarea difícil, bastante difícil, pero es una actividad necesaria cuya habilidad solo se adquiere con la experiencia.

El modelado es la actividad más delicada e importante en la realización de una aplicación con base de datos. Al igual que en el desarrollo de un sistema, toda modificación al esquema de base de datos debe realizarse primero en el modelo conceptual, no en el lógico ni en el físico. La habilidad de crear buenos modelos es una cualidad que se adquiere con la experiencia.

**Metas y beneficios**

* Registrar los requerimientos de datos de un proceso de negocio.
* Dicho proceso puede ser demasiado complejo y se tendrá que crear un "Enterprise data model", el cual deberá estar constituido de líneas individuales.
* Permite observar:
  + Patrones de datos
  + Usos potenciales de los datos

**2.Tipos de modelado de datos**

Un modelo de base de datos:es un tipo de modelo de datos que determina la estructura lógica de una [base de datos](http://es.wikipedia.org/wiki/Base_de_datos) y de manera fundamental determina el modo de almacenar, organizar y manipular los datos.

**Entre los modelos lógicos comunes para bases de datos se encuentran:**

* [Modelo jerárquico](http://es.wikipedia.org/wiki/Modelo_jer%C3%A1rquico)
* [Modelo en red](http://es.wikipedia.org/wiki/Base_de_datos_en_red)
* [Modelo relacional](http://es.wikipedia.org/wiki/Modelo_relacional)
* [Modelo entidad–relación](http://es.wikipedia.org/w/index.php?title=Modelo_entidad%E2%80%93relaci%C3%B3n&action=edit&redlink=1)
* Modelo entidad–relación extendido
* [modelo de objetos](http://es.wikipedia.org/wiki/Base_de_datos_orientada_a_objetos)
* [modelo documental](http://es.wikipedia.org/wiki/Base_de_datos_documental)
* Modelo entidad–atributo–valor
* [modelo en estrella](http://es.wikipedia.org/wiki/Esquema_en_estrella)

**Los modelos físicos de datos incluyen:**

* índice invertido
* fichero plano

**Otros modelos lógicos pueden ser:**

* modelo asociativo
* [modelo multidimensional](http://es.wikipedia.org/wiki/Base_de_datos_multidimensional)
* modelo multivalor
* modelo semántico
* [base de datos XML](http://es.wikipedia.org/wiki/Base_de_datos_XML)
* grafo etiquetado
* *Triplestore*

**Básicamente son 3:**

* Conceptual: muy general y abstracto, visión general del negocio/institución.
* Lógico: versión completa que incluye todos los detalles acerca de los datos.
* Físico: esquema que se implementara en un manejador de bases de datos (DBMS).

En las siguientes secciones se analizarán los aspectos relacionados con el modelado conceptual, más adelante y teniendo ya un modelo lógico se procederá a estudiar la representación física del mismo.

Es decir para construir un diseño de la base de datos se debe tener en cuenta lo siguiente:

**2.1Fases del diseño**

* Fase Inicial
* Diseño conceptual
* Implementación física

**Principios de diseño**

**Fidelidad:** se debe crear siempre un modelo que satisfaga las necesidades del problema, no sirve un modelo correcto si no cumple con la realidad que se pretende representar.

**Evitar redundancia:** una de las ventajas del diagrama e-r es que nos permite distinguir de una manera fácil y visual todos los entes y sus relaciones, de manera que es muy fácil identificar si un atributo se esta repitiendo en varias entidades o si una relación es innecesaria.

**Simplicidad:** siempre hay que procurar hacer el modelo tan simple como sea posible (sin olvidar la fidelidad) de manera que sea fácil de entender, fácil de extender y fácil de implementar.

**Escoger los elementos correctos**: es ocasiones es difícil identificar si una relación, elemento o atributo es correcto, para ello hay que analizar en perspectiva el diagrama y, por ejemplo si se observa una entidad con solo un atributo y que únicamente presenta relaciones de 1, entonces probablemente estamos hablando de un atributo y no de una entidad.

**Relaciones n-arias:**Aún cuando se pueden presentar casos en los que una relación terciaria o n-aria parezca más conveniente, es mejor siempre pensar en términos de relaciones binarias únicamente. En el peor de los casos de que exista una relación n-aria forzosa, lo que se debe hacer es convertir esa relacion R en entidad E y corregir todas las relaciones que tenía R de manera que ahora esa nueva entidad se relacione con todas las entidades que anteriormente esta.

**2.2Modelado de Datos Conceptual**

**Conceptos básicos**

**Algunos aspectos a considerar al momento de realizar el modelado/análisis**

* No pensar físicamente, pensar conceptualmente
* No pensar en procesos, pensar en estructura
* No pensar en navegación, pensar en términos de relaciones

**Modelos conceptuales**

**Existen distintos tipos de modelos conceptuales:**

**Basados en registros**

* Jerárquico: datos en registros, relacionados con apuntadores y organizados como colecciones de árboles
* Redes: datos en registros relacionados por apuntadores y organizados en gráficas arbitrarias
* Relacional: datos en tablas relacionados por el contenido de ciertas columnas

**Basados en objetos**

* Orientado a objetos: datos como instancias de objetos (incluyendo sus métodos)
* Entidad-relación: datos organizados en conjuntos interrelacionados de objetos (entidades) con atributos asociados

**Modelo Entidad-Relación**

**¿Qué es el modelo entidad-relación?**

Es uno de los varios modelos conceptuales existentes para el diseño de bases de datos. Fue inventado por Peter Chen, en los años setenta. El propósito de este modelo es simplificar el diseño de bases de datos a partir de descripciones textuales de los requerimientos.

Generalmente todo modelo tiene una representación gráfica, para el caso de datos el modelo más popular es el modelo entidad-relación o diagrama E/R.

Se denomina así debido a que precisamente permite representar relaciones entre entidades (objetivo del modelado de datos).

**El modelo debe estar compuesto por:**

* Entidades
* Atributos
* Relaciones
* Cardinalidad
* Llaves

**Conjuntos de entidades y atributos**

* Entidades: todo lo que existe y es capaz de ser descrito (sustantivo).
* Atributos: es una característica (adjetivo) de una entidad que puede hacer 1 de tres cosas:
  + Identificar
  + Relacionar
  + Describir

**En el diseño se pueden considerar 3 categorías de atributos**

* Simples o compuestos: ya sea que el atributo sea un todo o bien este compuesto
  + Color es simple, toma valores rojo, azul, etc
  + Nombre es compuesto, contiene nombre de pila, apellido materno, apellido materno
* Con valores simples o multivaluados: en base a si consisten de un solo valor o un conjunto de valores.
  + Teléfono o Teléfonos
* Derivados: que se pueden calcular en base a otros atributos
  + El promedio de préstamos se puede derivar si tenemos los valores de cada préstamo realizado a un persona

**NOTA: en la práctica es mejor considerar "siempre" a todos los atributos como simples y con valores simples**

**Llaves**

* Súper llave: conjunto de uno o más atributos que "juntos" identifican de manera única a una entidad
* Llave candidata: es una super llave mínima
* Llave primaria: la seleccionada para identificar a los elementos de un conjunto de entidades.

**Ejemplo:**

Teniendo los atributos de la entidad "persona"

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Dirección | Teléfono | CURP |

* + Las súperllaves serían:
    - Nombre y Dirección
    - Nombre y CURP
    - CURP
  + Las llaves candidatas serían
    - Nombre y Dirección
    - CURP
  + La llave primaria sería
    - CURP

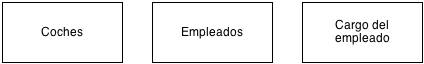
**Elementos del modelo entidad-relación**

**(Entidad, Atributos, Relación)**

**Entidad:** Las entidades representan *cosas* u *objetos* (ya sean reales o abstractos), que se diferencian claramente entre sí.

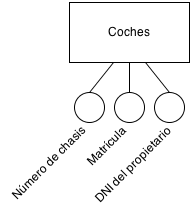
* Coches (objeto físico): contiene la información de cada taller.
* Empleado (*objeto* físico): información de los trabajadores.
* Cargo del empleado (*cosa* abstracta): información de la función del empleado.

Estas entidades se representan en un diagrama con un rectángulo.



**Atributos:**Los atributos definen o identifican las características de entidad (es el contenido de esta entidad). Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha…).

Tomando en cuenta el ejemplo anterior unos posibles atributos serían los siguientes: *número de chasis*, *matrícula*, *DNI del propietario*, *marca*, *modelo entre otros.*

Los atributos se representan como círculos que descienden de una entidad, y no es necesario representarlos todos, sino los más significativos,

En un modelo relacional (ya implementado en una base de datos) un ejemplo de tabla dentro de una *BBDD* podría ser el siguiente.

|  |  |  |
| --- | --- | --- |
| Número de chasis | Matrícula | DNI del propietario |
| 5tfem5f10ax007210 | 4817 BFK | 45338600L |
| 6hsen2j98as001982 | 8810 CLM | 02405068K |
| 5rgsb7a19js001982 | 0019 GGL | 40588860J |

Este ejemplo es con tres atributos, pero un coche o entidad podría tener cientos (si fuese necesario) y seguirían la misma estructura de columnas, tras implementarlo en una *BBDD*.

**Relación:** Es un vínculo que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable.

Ejemplo:Los empleados del taller (de la entidad “Empleados“) tienen un cargo (según la entidad “Cargo del empleado“). Es decir, un atributo de la entidad “*Empleados* “especificará que cargo tiene en el taller, y tiene que ser idéntico al que ya existe en la entidad “*Cargo del empleado*“.

Las relaciones se muestran en los diagramas como rombos, que se unen a las entidades mediante líneas.



**Empleados**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **DNI** | **Cargo** |
| Carlos Sánchez | 45338600L | 001 |
| Pepe Sánchez | 02405068K | 002 |
| Juan Sánchez | 40588860J | 002 |

**Cargo del empleado**

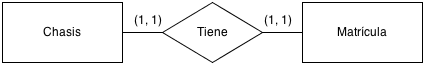
|  |  |
| --- | --- |
| **ID del cargo** | **Descripción** |
| 001 | Jefe de taller |
| 002 | Mecánico |

**Relaciones de cardinalidad o Relación de Modelos de Datos.**

Podemos encontrar distintos tipos de relaciones según como participen en ellas las entidades.

Es decir cada empleado puede tener un cargo, pero un mismo cargo lo pueden compartir varios empleados.Cada relación tiene un nombre que describe su función. Las relaciones se representa gráficamente mediante un rombo etiquetado en su interior con un verbo, este rombo se debe unir mediante líneas con las entidades (rectángulos) que relaciona y especifica cuantos *objetos* o *cosas* (de cada entidad) pueden intervenir en esa relación.Los indicadores numéricos indican el primero el número mínimo de registros en una relación y posteriormente el máximo (si no hay límite se representa con una “*n*“).

**Uno a uno:** Una entidad se relaciona únicamente con otra y viceversa.



Ejemplo: si tuviésemos una entidad con distintos chasis y otra con matrículas deberíamos de determinar que cada chasis solo puede tener una matrícula (y cada matrícula un chasis, no más en ningún caso).

**Uno a varios o varios a uno:** determina que un registro de una entidad puede estar relacionado con varios de otra entidad, pero en esta entidad existir solo una vez.

**Varios a varios:** determina que una entidad puede relacionarse con otra con ninguno o varios registros y viceversa.

Ejemplo. En el taller un coche puede ser reparado por varios mecánicos distintos y esos mecánicos pueden reparar varios coches distintos.

Esto es para hacerle una pregunta a la profesora

Uno a Uno.En una relación uno a uno, cada registro de la Tabla A sólo puede tener un registro coincidente en la Tabla B y viceversa.

Uno a Muchos.La relación uno a varios es el tipo de relación más común. En este tipo de relación, un registro de la Tabla A puede tener muchos registros coincidentes en la Tabla B, pero un registro de la Tabla B sólo tiene un registro coincidente en la Tabla A.

Muchos a Muchos.En una relación varios a varios, un registro de la Tabla A puede tener muchos registros coincidentes en la Tabla B y viceversa.

Diagrama E-R

|  |
| --- |
| http://ict.udlap.mx/people/carlos/is341/images/e-r_notation.png |

Notación empleada para elaborar modelos E-R

**Guías de nombramiento**

Es importante mantener guías o reglas para poder tener una documentación uniforme y consistente de todos los datos.

* Entidades: una sola palabra (en singular) y con mayúsculas
* Atributos:
  + FirstName
  + first\_name
  + de relacion: VendorID, ProductName
* Valores: definir que valores son válidos (NULL no es un valor)

**Cardinalidades**

En base al número de instancias involucradas en cada relación, éstas presentan un cardinalidad, que puede ser:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | http://ict.udlap.mx/people/carlos/is341/images/modelo8.gif | (Muchos a Muchos) | | http://ict.udlap.mx/people/carlos/is341/images/modelo9.gif | (Uno a Muchos) | | http://ict.udlap.mx/people/carlos/is341/images/modelo10.gif | (Uno a Uno) | |